

Team Average Joes



2019-11-05

SEG3102

Professor Stéphane S. Somé

Christopher Francis 8661478

Jaimin Patel 8721083

Samuel Worrod 8653389

Jordan Benoit 8346159

Deliverable 2, Due 2019-11-11

Table of Contents

1.0 Executive Summary	3
2.0 Team Roles	3
3.0 Assumptions	4
4.0 UML Domain Model (Class Diagram)	4
5.0 System Responsibilities	4
Register User	4
Instructor Log In	5
Set Up Parameters	5
Modify Parameters	5
Visualize Students teams	5
Instructor Log Out	5
Student Log In	6
Create Team	6
Join Team	6
Accept New Students	7
Student Log Out	7
6.0 Use Case Realizations	8
Register User	8
Instructor Log In	8
Set Up Parameters	9
Modify Parameters	9
Visualize Teams	10
Instructor Log Out	10
Student Log In	11
Create Team	12
Join Team	12
Accept New Students	13
Student Log Out	13
7.0 Analysis Level Class Diagram	14
8.0 References	14

Table of Figures

Figure Number	Figure Name	Page Number
0	<i>UML Domain Model</i>	4
1	<i>Use Case Realization for Register User</i>	8
2	<i>Use Case Realization for Instructor Log In</i>	8
3	<i>Use Case Realization for Setup Parameters</i>	9
4	<i>Use Case Realization for Modify Parameters</i>	9
5	<i>Use Case Realization for Visualize Teams</i>	10
6	<i>Use Case Realization for Instructor Log Out</i>	10
7	<i>Use Case Realization for Student Log In</i>	11
8	<i>Use case Realization for Create Team</i>	12
9	<i>Use Case Realization for Join Team</i>	12
10	<i>Use Case Realization for Accept New Students</i>	13
15	<i>Use Case Realization for Student Log Out</i>	13
16	<i>Analysis Class Diagram</i>	14

Table of Tables

Table Name	Page Number
Team Roles	3

1.0 Executive Summary

The purpose of this document is to provide our team's analysis level and use case realizations of the Team Management System (TMS). Included in the document are the division of team roles, assumptions and design decisions about the system, the UML domain model, use case contract descriptions, use case realizations, and an analysis level UML class diagram.

2.0 Team Roles

Member Name	Responsibilities
Samuel Worrod	<ul style="list-style-type: none">● 5.0 System Responsibilities<ul style="list-style-type: none">○ 9. Instructor Log Out○ 10. Student Log In○ 11. Create Team● 6.0 Use Case Realizations<ul style="list-style-type: none">○ 6. Instructor Log Out○ 7. Student Log In○ 8. Create Team
Christopher Francis	<ul style="list-style-type: none">● 4.0 UML Domain Model● 5.0 System Responsibilities<ul style="list-style-type: none">○ 13. Join Team○ 14. Accept New Students○ 15. Student Log Out● 6.0 Use Case Realizations<ul style="list-style-type: none">○ 13. Join Team○ 14. Accept New Students○ 15. Student Log Out● 7.0 Analysis Class Diagram
Jordan Benoit	<ul style="list-style-type: none">● 5.0 System Responsibilities<ul style="list-style-type: none">○ 1. Register User○ 2. Instructor Log In○ 3. Set up Parameters● 6.0 Use Case Realizations<ul style="list-style-type: none">○ 1. Register User○ 2. Instructor Log In○ 3. Set up Parameters
Jaimin Patel	<ul style="list-style-type: none">● 5.0 System Responsibilities<ul style="list-style-type: none">○ 5. Visualize Student Teams

	<ul style="list-style-type: none"> 6.0 Use Case Realizations <ul style="list-style-type: none"> 5. Visualize Student Teams
Everyone	<ul style="list-style-type: none"> Tables <ul style="list-style-type: none"> Table of figures Table of Team Roles Assumptions Analysis Class Diagram Brainstorming

Table 1: Team Roles

3.0 Assumptions

4.0 UML Domain Model (Class Diagram)

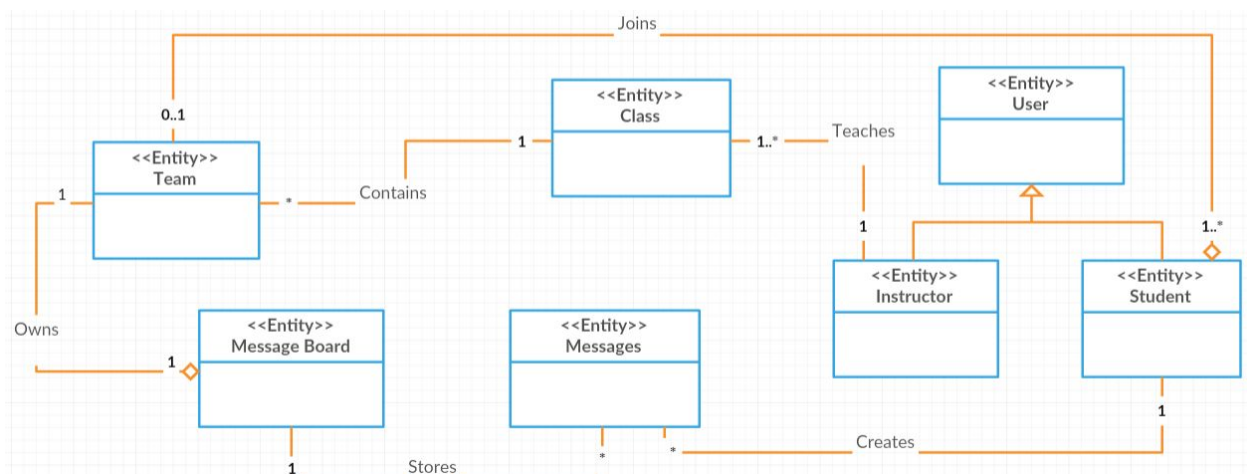


Figure 0: UML Domain Model

5.0 System Responsibilities

Register User

Name: //registerUser(userInfo)

Responsibilities: creates a new user in the TMS System

Cross References: Use Cases: Register User

Preconditions:

- User must have a user identification of type instructor or student

Postconditions: A new (Instructor/Student) user is registered

Instructor Log In

Name: //sign_in(loginInfo)

Cross References: Use Cases: Instructor Log In

Preconditions:

- Instructor who is logging in must be a registered user

Postconditions: none

Set Up Parameters

Name: //setup_team_parameters()

Responsibilities: Creates an instance of 'Team Creation' with the set parameters

Cross References: Use Cases: Set Up Parameters

Preconditions:

- User must be of type *Instructor*

Postconditions: The system will allow students to create teams with the given parameters

Modify Parameters

Name: //modify_team_parameters()

Responsibilities: Edits the set parameters of team creation

Cross References: Use Cases: Modify Parameters

Preconditions:

- An instance of 'Team Creation' must already exist
- User must be of type *Instructor*

Postconditions: The existing 'Team Creation' has been modified and updated

Visualize Students teams

Name: //visualizeTeams()

Responsibilities: Visualize all *teams* in an *instructor's class*

Cross References: Use Cases: Visualize Students Teams

Preconditions:

- *User Account* must be of type *Instructor*
- Instructor must have a non-null instance of *Class* class
- Class must have at least one non-null instance of *Team* team

Postconditions:

- The *system* displays list of all *teams*

Instructor Log Out

Name: //instructorLogOut(instructor)

Cross References: Use Cases: Instructor log out

Preconditions:

- *instructor* is a non null instance of *User Account*
- *instructor.loggedin* is *true*

Postconditions:

- *instructor.loggedin* is *false*

Student Log In

Name: //sign_in(loginInfo)

Cross References: Use Cases: Student Log In

Preconditions:

- Student who is logging in must be a registered user

Postconditions:

- *student.loggedin* is *true*

Create Team

Name: //createTeam(student, class)

Responsibilities: create a *team* for a *class*

Cross References: Use Cases: Create Team

Preconditions:

- There is an existing instance of *Class* class
- *User Account* must be of type *student*
- Student must have a non-null instance of *Class* class

Postconditions:

- A new instance of *Team* *team* was created (instance creation)

Join Team

Name: //joinTeam(student, team)

Responsibilities: create a request for *student* to join *team*

Cross References: Use Cases: Join Team

Preconditions:

- There is an existing instance of *Team* *team*
- *Student* is a non null instance of *User Account*

Postconditions:

- A new instance of *Request* *request* was created (instance creation)

Accept New Students

Name: //acceptStudent(student, team)

Cross References: Use Cases: Accept new students

Preconditions:

- *team* is a non null instance of *Team*
- *student* is a non null instance of *User Account*
- Logged in *user* of instance *User Account* has attribute *isLiaison* set to *true*
- *team.maximum* has not been reached.

Postconditions:

- The attribute *team.maximum* has been increased by 1. (attribute modification)
- *student* was associated with *team*. (association creation)

Student Log Out

Name: //studentLogOut(student)

Cross References: Use Cases: Student log out

Preconditions:

- *student* is a non null instance of *User Account*
- *student.loggedin* is *true*

Postconditions:

- *student.loggedin* is *false*

6.0 Use Case Realizations

Register User

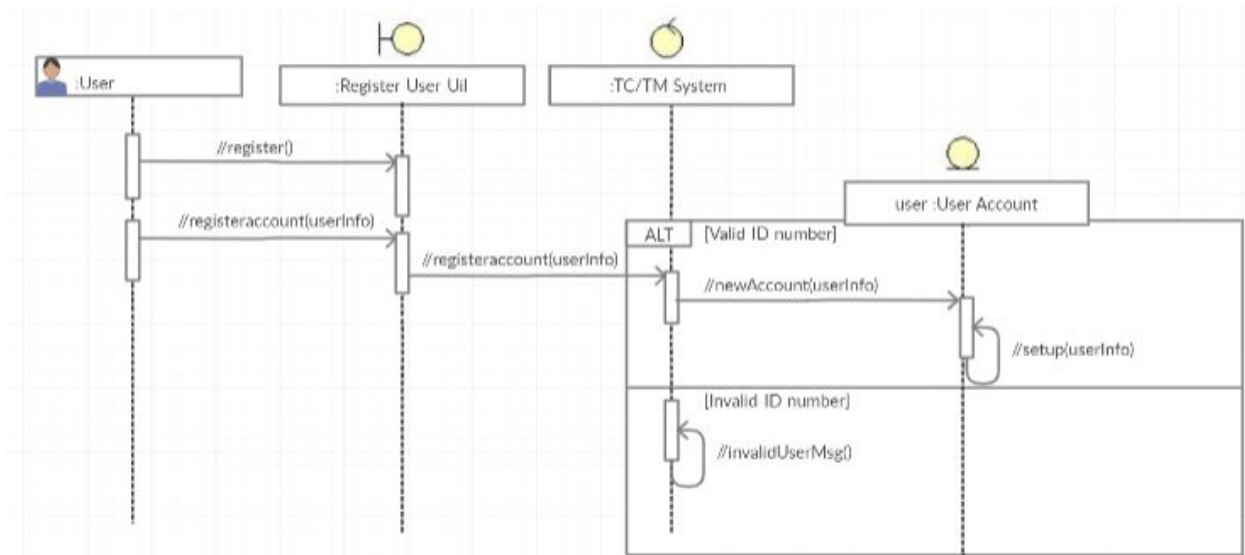


Figure 1: Use Case Realization for Register User

Instructor Log In

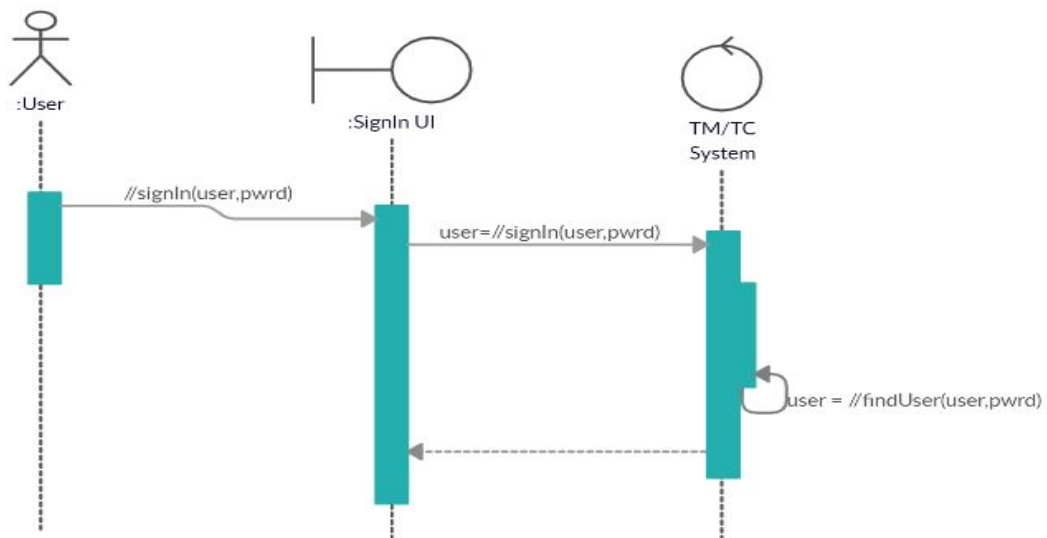


Figure 2: Use Case Realization for Instructor Log In

Set Up Parameters

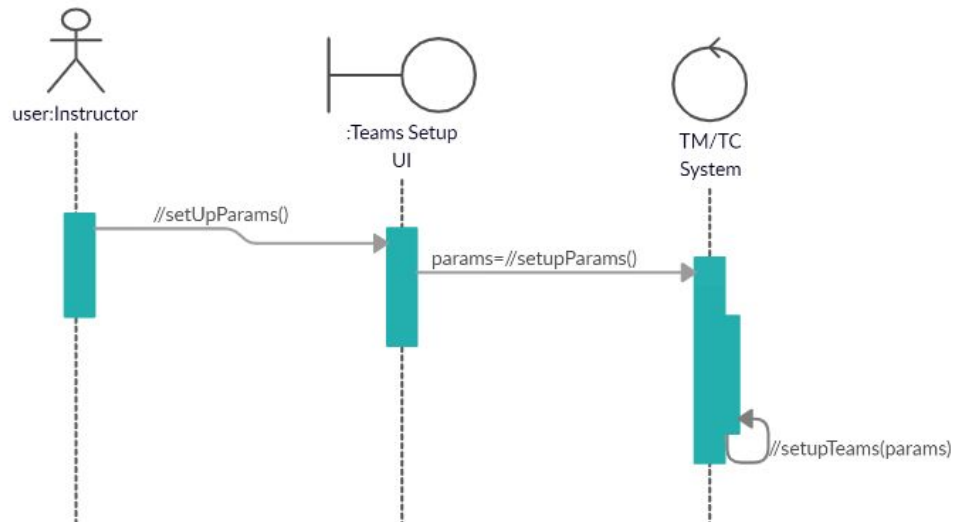


Figure 3: Use Case Realization for Set Up Parameters

Modify Parameters

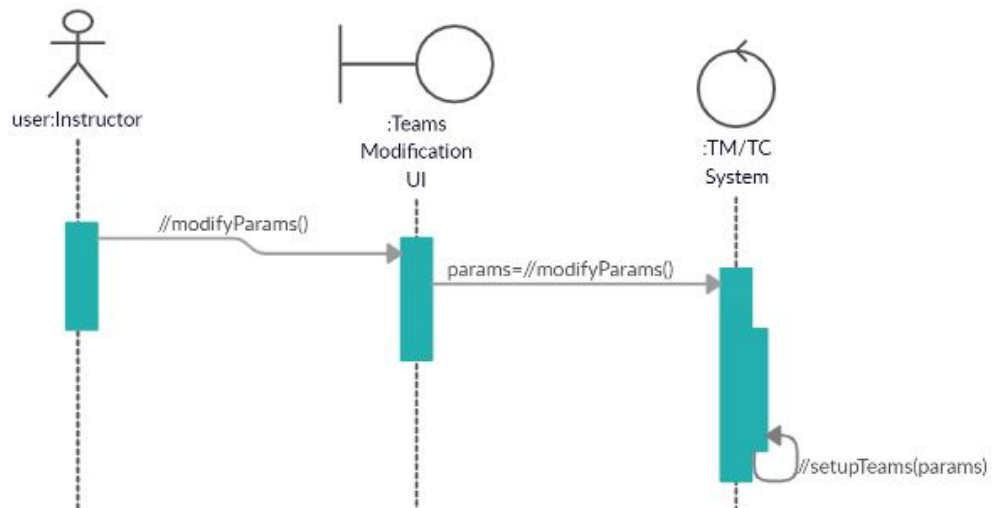


Figure 4: Use Case Realization for Modify Parameters

Visualize Teams

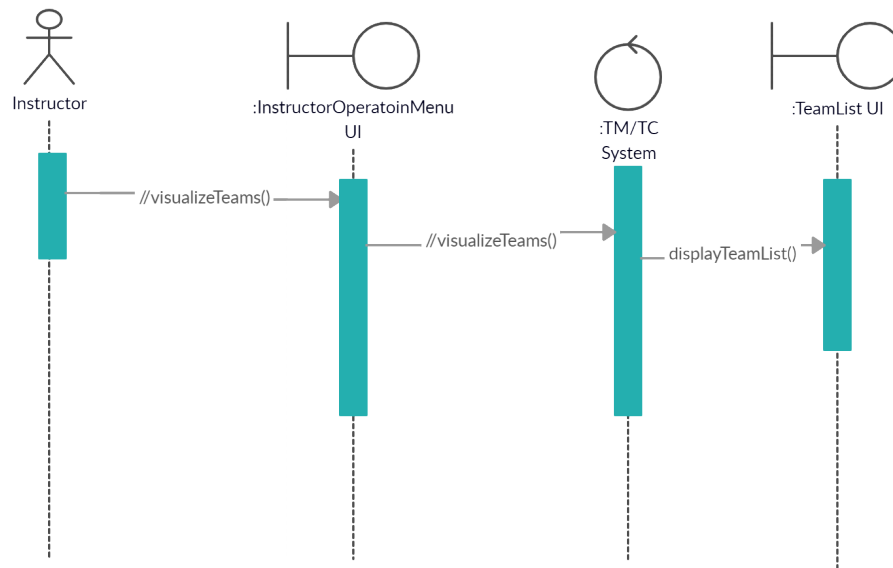


Figure 5: Use Case Realization for Visualize Teams

Instructor Log Out

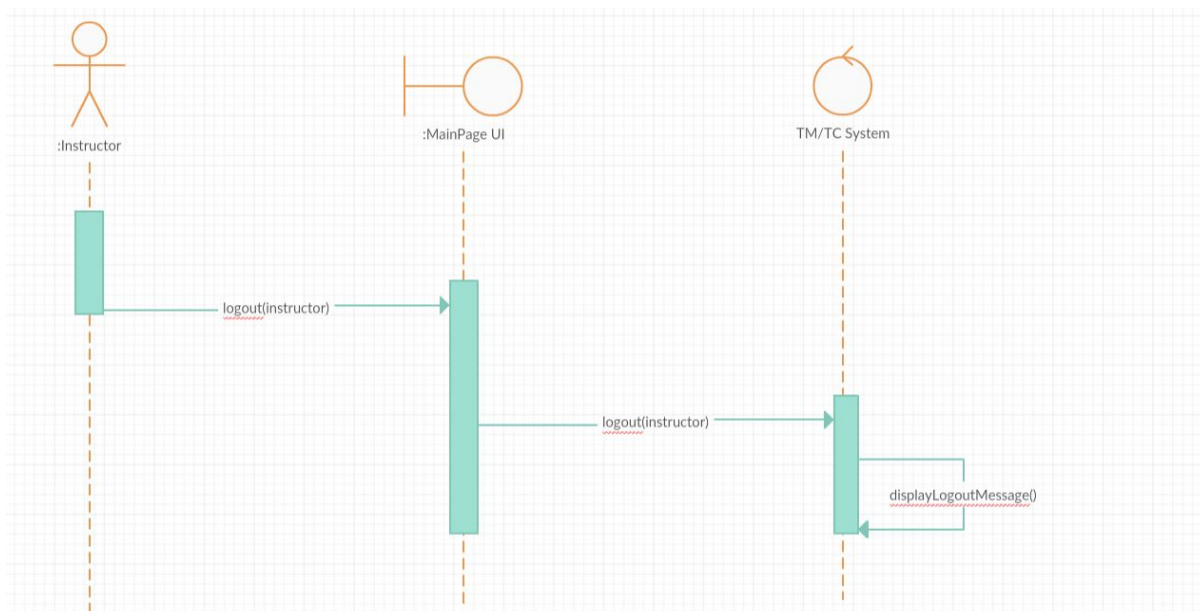


Figure 6: Use Case Realization for Instructor Log Out

Student Log In

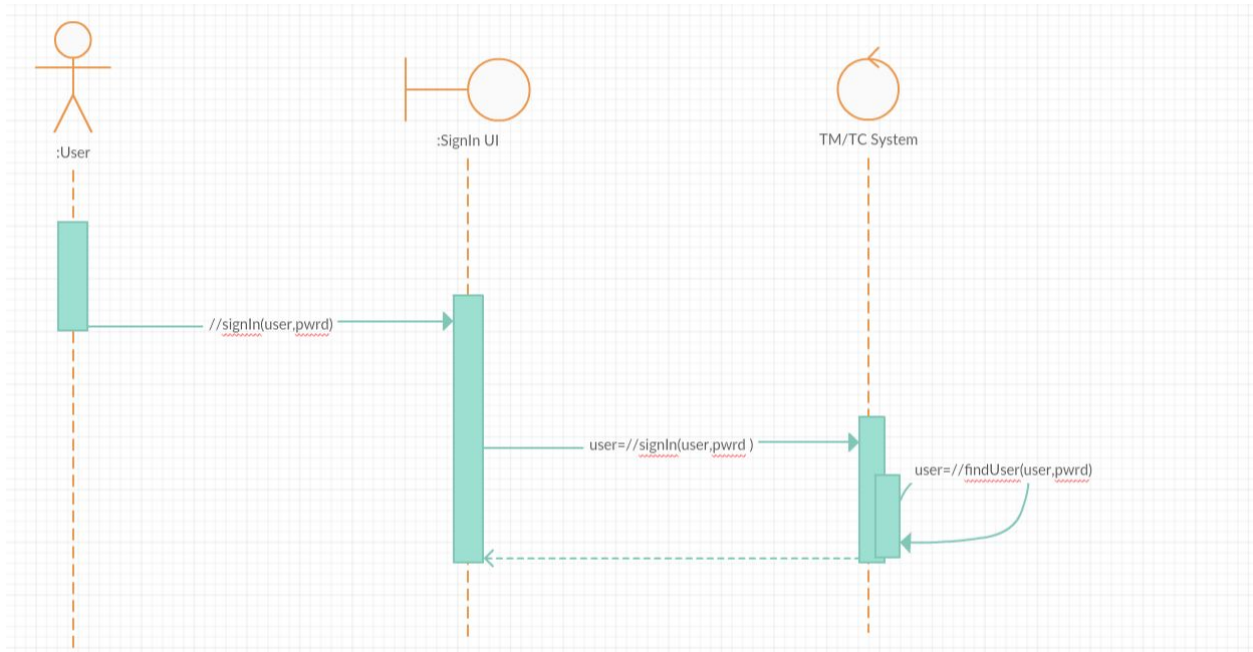


Figure 7: Use Case Realization for Student Log In

Create Team

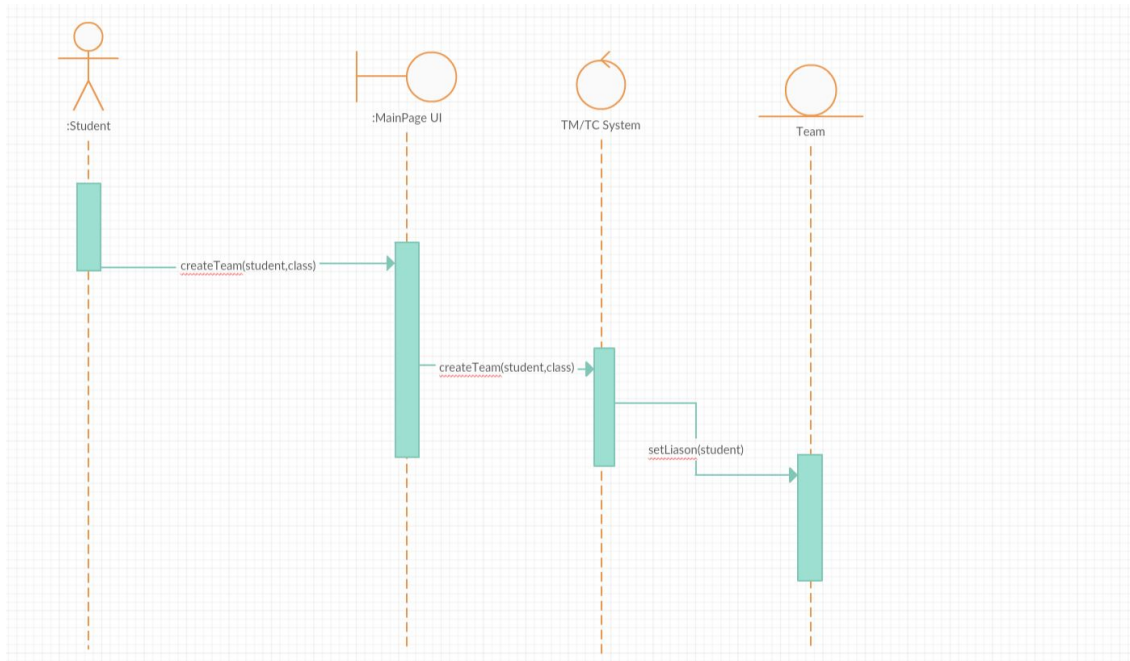


Figure 8: Use Case Realization for Create Team

Join Team

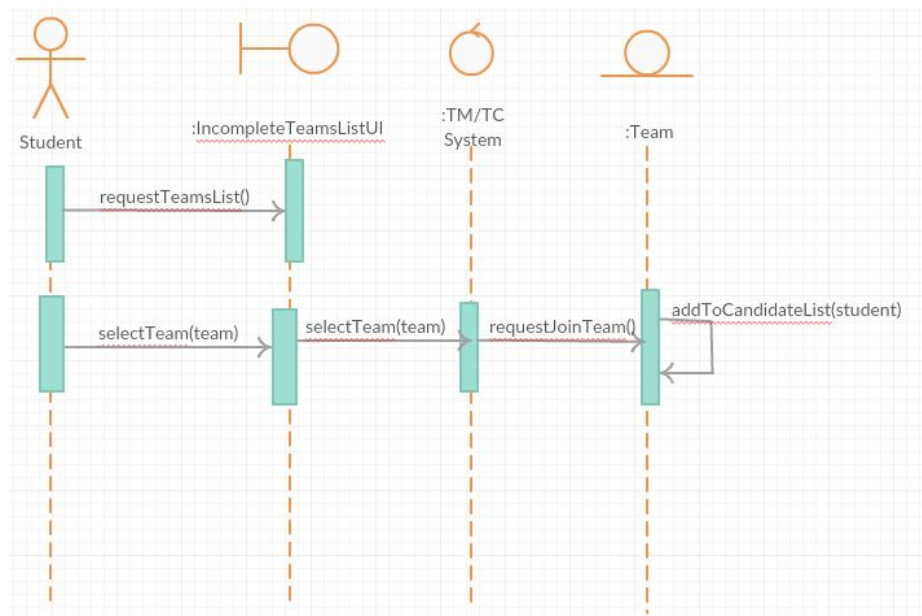


Figure 9: Use Case Realization for Join Team

Accept New Students

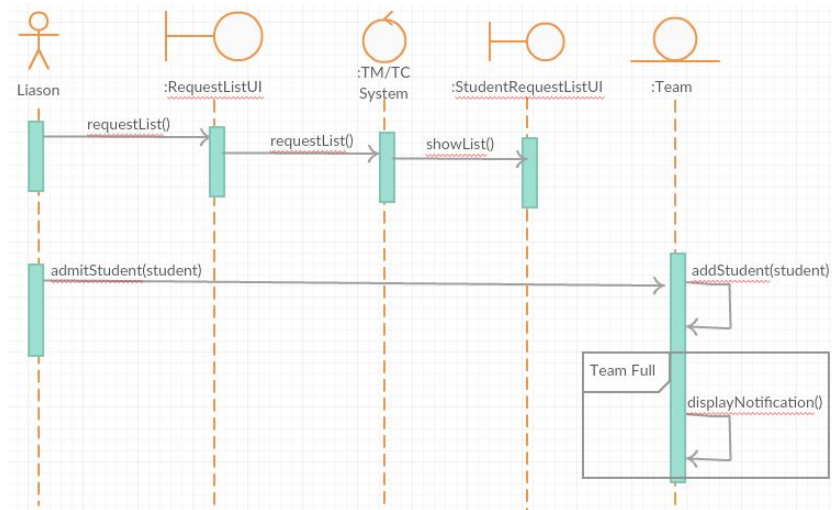


Figure 10: Use Case Realization for Accept New Students

Student Log Out

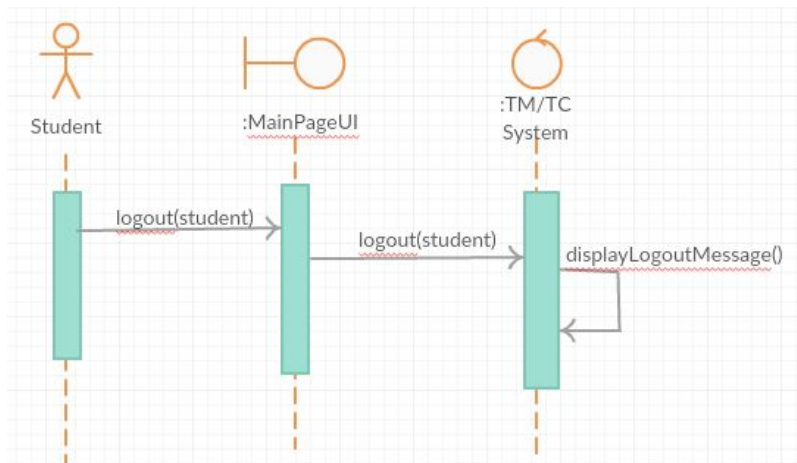


Figure 11: Use Case Realization for Student Log Out

7.0 Analysis Level Class Diagram

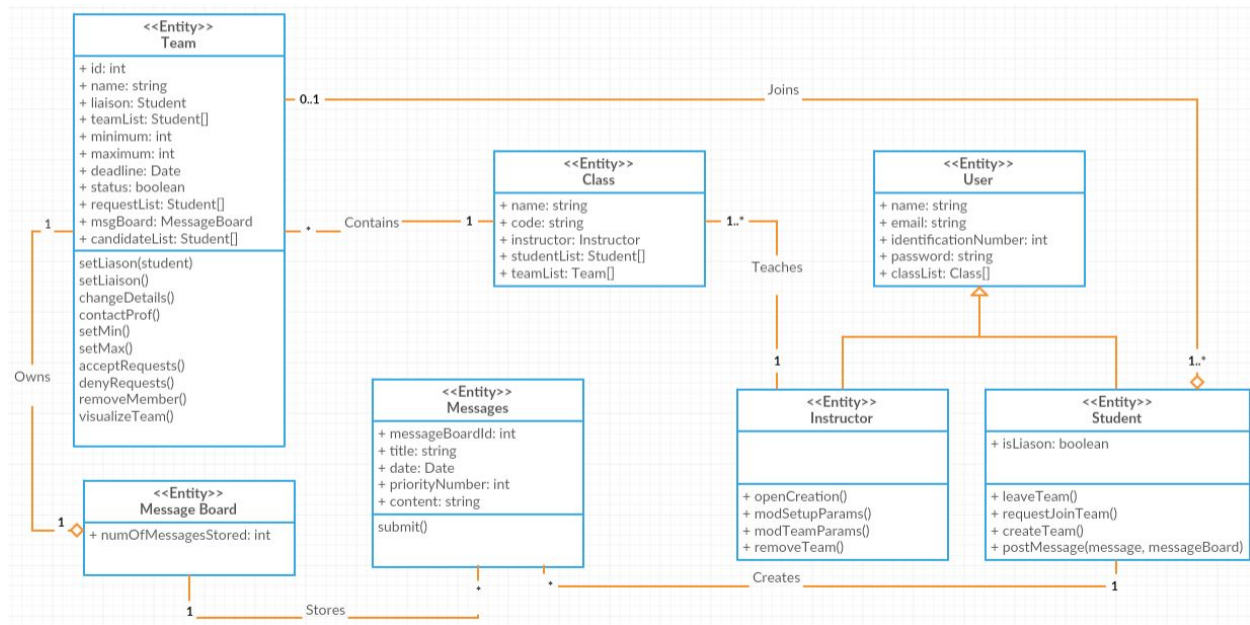


Figure 12: Analysis Class Diagram

8.0 References

Not applicable for this document.